**PIVOT TABLE**

PIVOT TABLE, ממש כמו באקסל, זו טבלת ציר. בה שורות הופכות לעמודות ולהפך.

כאנליסטים יוצא לנו לא מעט להשתמש בפעולה הזאת, שמאפשרת לנו להצליב בין נתונים ולחקור בצורה נוחה יותר את המידע שלנו. למשל, אם נרצה לחקור כמה כסף נכנס מכל עיר גם לכל חברת שילוח וגם לכל ספק, נצטרך לבצע קיבוץ לשלושת הפרמטרים, וכאשר נעשה זאת נקבל טבלה מאוד ארוכה שבעזרת ORDER BY נוכל רק להתחיל לחפש את הנתונים שאנחנו מחפשים. בפועל זו לא שיטה יעילה או ברורה בכלל.

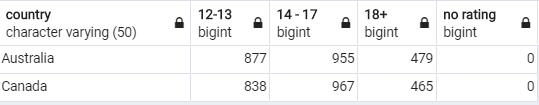
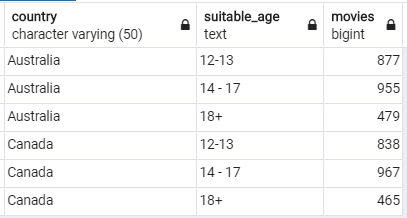
PIVOT TABLE נותנת לנו את האופציה להציג עמודה לספק, עמודה לחברת שילוח ועמודה לכל עיר שערכיה סכומי הכסף, אפשרות שמפשטת את תהליך החקירה בצורה משמעותית.

אז איך נעשה דבר כזה?

טבלת ציר בנויה מכמה חלקים: פיבוטינג, שורות סיכום, עמודת תיאור ברורה ועמודת סיכום.

**PIVOTING**

החלק הראשון והפשוט מבניהם הוא פעולת **הפיבוטינג** עצמה שלמדתם בשיעור. אנחנו נתמקד בדוגמה של הקטגוריה הספציפית 12-13.

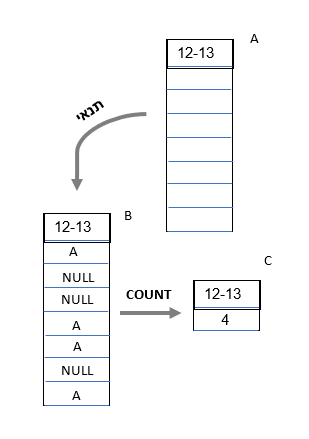


בשביל להפוך שורה (קטגוריה כלשהי מתוך עמודה) לעמודה, כמו שלמדתם, ניצור קודם כל עמודה חדשה, בעזרת CASE וניתן לה את השם של אותה הקטגוריה – השורה (A).

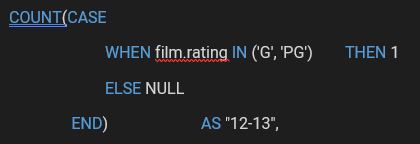
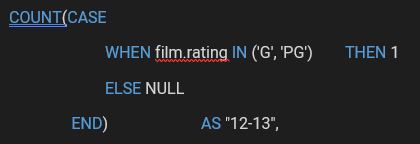
נבנה את העמודה לפי תנאי אחד שאומר שאם הקטגוריה הזאת מופיעה, יהיה תו כלשהו בעמודה. ברירת המחדל בעבור התנאים היא שאם אותה השורה היא לא מאותה הקטגוריה, יופיע הערך NULL וכך נבנית העמודה המותנית. (B).

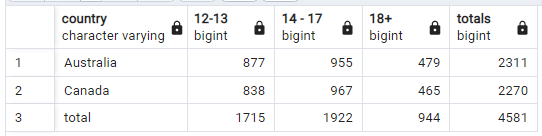
על העמודה נציב אגרגציה (כל אגרגציה לא רק ספירה) וזהו, זה התהליך להפיכת קטגוריה לעמודה. (C).

נבנה עמודה מותנית בעבור כל קטגוריה וכך נקבל טבלת ציר.



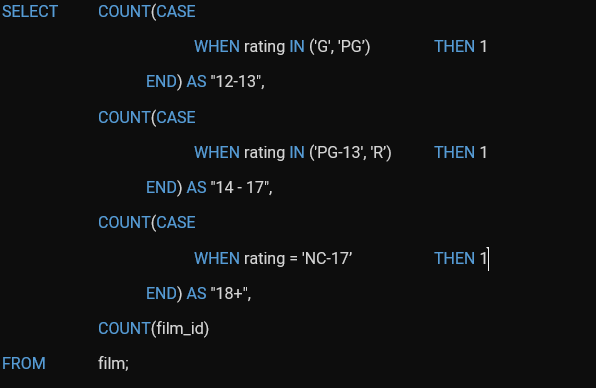
אגרגציה



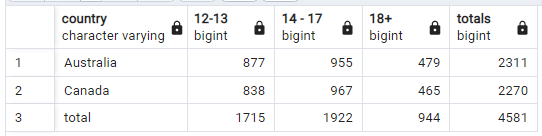


**עמודת סיכום**

מאפיין נוסף וחשוב לטבלת ציר הוא **עמודת סיכום**. כרגע, כל עמודה שלנו מורכבת מאגרגציה שקיבצנו לפי איזושהי עמודה שיצרנו לפי איזושהי קטגוריה. למשל בדוגמה הקודמת, העמודה "12-13" היא עמודה שמקובצת לפי עמודת המדינות ונוצרה לפי קטגורית הגיל המתאים. ככה אנחנו חוקרים את הצטלבות המידע בין שני המאפיינים. כדי לחקור את המידע עד סופו לטבלאות ציר לרוב תהיה עמודת סיכום שמתייחסת רק לפרמטר האופקי (או שורת סיכום שנגיע אליה בהמשך). אם לא נרצה להתייחס לפרמטר האנכי פשוט ניצור את אותה האגרגציה אבל הפעם לא על עמודה מותנית. למשל אם ספרנו עד עכשיו רק סרטים בהתאמת גיל מסוימת, הפעם נספור את כל הסרטים.



טבלת התוצאה:



**ROLLUP**

כמו שציינו, חוץ מעמודת סיכום נרצה ליצור גם **שורת סיכום**. כאן התהליך קצת מסתבך. כדי להתייחס רק לפרמטר האופקי (המדינות) יצרנו את אותה האגרגציה אבל ללא עמודה מותנית. בשביל להתייחס רק לפרמטר האנכי (קטגוריות הגיל) נצטרך ליצור שורה שלא מתייחסת לקיבוץ או יותר מדויק, מסכמת את הקיבוץ שלנו. בשביל זה בדיוק יש לנו את הפונקציה ROLLUP שהיא מעין הרחבה של הפסוקית GROUP BY.

מבנה הכתיבה:

SELECT col1, col2,

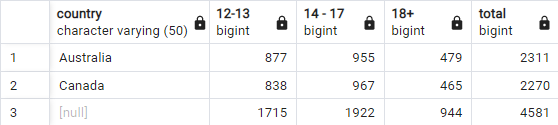
Aggregate\_function(col3)

FROM table1

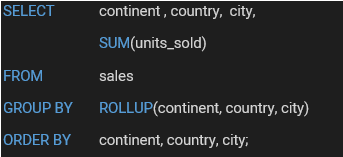
GROUP BY ROLLUP(col1, col2)

במקרה שלנו נוסיף ROLLUP לקיבוץ לפי מדינות וכבר כך נקבל שורת סיכום בתחתית הטבלה שבה לא מופיעה אף מדינה ומסוכמת כמות הסרטים בכל קטגוריה.

(מומלץ להריץ את הדוגמאות ב-PGAdmin)

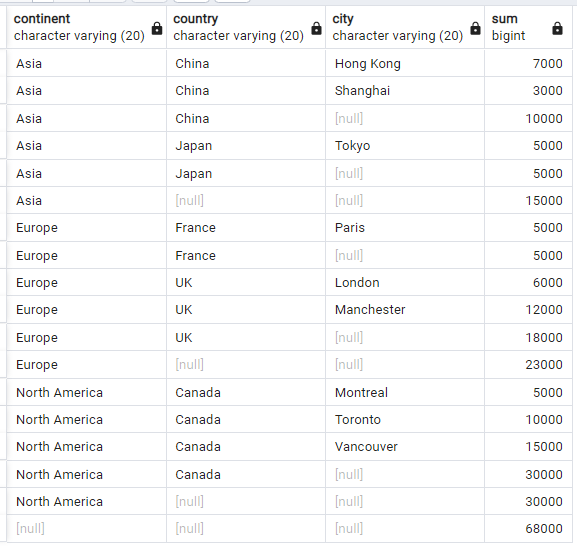


ROLLUP היא פונקציה היררכית (כמו ORDER BY) שיכולה לפעול על יותר מרק עמודה אחת בקיבוץ. לרוב נשתמש בה בדברים היררכיים כמו שנים-חודשים-ימים או מדינות-מחוזות-ערים. ממש כמו ב-ORDER BY, נסדר את העמודות לפי סדר ההיררכיה שלהן. משמאל לימין, מהגדול לקטן. ככה המנוע ידע לפי מה לסכם לנו את השורות. נשלוף הפעם ממסד נתונים לא מוכר שמציג מידע על כמות מכירות בכל עיר ואת המדינה והיבשת שלה (רק לצורך ההדגמה). ננסה לשלוף את כמות המכירות לכל יבשת, לכל מדינה ולכל עיר (משמאל לימין, גדול לקטן):



חשוב ORDER BY כדי שהשליפה תהיה קריאה!

טבלת התוצאה שלנו:



אפשר לראות שלכל עיר יש את הסכום שלה בעמודת הסכום, לכל מדינה יש שורה שבה אין ערך בעמודת הערים (חץ כחול) שהערך בה הוא סכום המכירות לאותה מדינה וככה גם לכל יבשת יש שורת סיכום (חץ ירוק) ובתחתית הטבלה נוספת שורת סיכום אחרונה שנותנת לנו את סכום המכירות הכולל של כל היבשות כל המדינות וכל הערים ביחד (חץ סגול).

**הערה חשובה:**

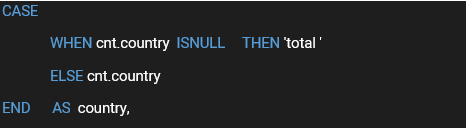
ב- PostgreSQL שורת ה- ROLLUP נועדה לאפשר הצגת שורות סיכום (לכל אגרגציה לא רק סכום – גם ממוצע למשל יכל להיסתכם לכל הקטגוריה), ללא הצגת נתונים מפורטים לצד הנתונים המצטברים. משום כך לא קיימת אפשרות להעניק שמות משמעותיים לעמודות ב-SELECT ובמקביל להשתמש בעמודת המפתח הראשי בהגדרת שורת הסיכום, כפי שניתן לעשות בפסוקית GROUP BY רגילה. ולכן, כאשר משתמשים ב- ROLLUP בלבד, ההנחיה היא **לקבץ לפי עמודות השם ולא לפי המפתח הראשי** על מנת לא לפגוע בקריאת טבלת התוצאה. לסיכום, נשתמש ב- ROLLUP כדי להוסיף שורות סיכום לאגרגציה שלנו. היא תוספת לפסוקית ה- GROUP BY שפועלת בצורה היררכית והיא המקום היחיד בו נקבץ לפי שם העמודה ולא לפי מפתח ראשי.

**PIVOT & CASE**

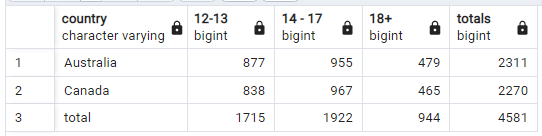
אז מה נשאר בשביל ליצור את טבלת הPIVOT שלנו? קריאות. קצת סטנדרטים. ממש כמו שלא נציג ללקוח עמודה עם עמודות לא ברורות, לא נציג ללקוח את עמודת הסיכום כשבתחילתה רק NULL.

אז מה הפתרון? CASE!

את ה- ROLLUP נבצע אמנם על עמודת המדינות אבל במקום להציג רק את עמודת המדינות בשליפה שלנו, נציג עמודת מדינות משופצת. ניצור עמודה מותנית שהתנאי היחיד שלה הוא שכשיש NULL, תירשם המילה total ובכל מצב אחר, עמודת המדינות כרגיל.



טבלת התוצאה:



כך נחליף את הNULL במלל ברור.

**יעילות השימוש ב- PIVOT TABLE**

כפי שציינו בהתחלה, PIVOT TABLE עוזר לפשט את תהליך החקירה. אבל איך? למה שנעדיף להשתמש ב PIVOT TABLE במקום בשליפה רגילה?

**יעילות שורות הסיכום -** שורות הסיכום נוחות לשימוש, אך האלטרנטיבה המוכרת יותר היא שימוש רגיל בפסוקית ה- GROUP BY. על אף שהן עושות את אותה פעולה (סכימת הנתונים) בדרכי תצוגה שונות, סכימה באמצעות ROLLUP תדרוש הרבה פחות משאבים מקיבוץ. למשל, אולי נתקלתם בפרויקט דאוס בצורך לקבץ את המידע על פי מספר עמודות – (לדוגמה – שעה, דקה ושנייה) בעבור קיבוץ כל העמודות והפעלת הפעולה הרצויה, נוצרת טבלת תוצאה גדולה מאוד שמציגה את כל שילובי הערכים האפשריים. יצירה של טבלה כזו דורשת משאבים רבים. לעומת זאת, ROLLUP רץ בצורה שונה מאחורי הקלעים שגורמת לסכימה להתבצע בצורה יעילה יותר. וכאשר מדובר במידע רב, הבדלי הריצה יכולים להיות משמעותיים מאוד.

לסיכום, יצירת PIVOT TABLE, שלמדתם ב-ל.ע כרוכה בביצוע כמה פעולות שונות, מעט מורכבות ומאוד שימושיות. וכעת למדתם כיצד כולן משתלבות ביחד למימוש טבלת ציר אחת.

תרגלו את מה שלמדתם ואל תפחדו לשאול שאלות אם משהו אינו מובן, בהצלחה!